



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/724,254	12/01/2003	Georg Biehler	Q78306	1606
23373	7590	08/22/2007	EXAMINER	
SUGHRUE MION, PLLC 2100 PENNSYLVANIA AVENUE, N.W. SUITE 800 WASHINGTON, DC 20037			WANG, BEN C	
		ART UNIT	PAPER NUMBER	
		2192		
		MAIL DATE	DELIVERY MODE	
		08/22/2007	PAPER	

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

S

Office Action Summary	Application No.	Applicant(s)
	10/724,254	BIEHLER ET AL.
	Examiner	Art Unit
	Ben C. Wang	2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 08 June 2007.
 2a) This action is **FINAL**. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-44 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-44 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413) Paper No(s)/Mail Date _____
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) <input type="checkbox"/> Notice of Informal Patent Application
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____	6) <input type="checkbox"/> Other: _____

DETAILED ACTION

1. Applicant's amendment dated June 8, 2007, responding to the Office action mailed February 8, 2007 provided in the rejection of claims 1-44, wherein claims 2-18, 20-26 and 28-44 are remained as original, and claim 1, 19, and 27 are amended.

Claims 1-44 remain pending in the application and which have been fully considered by the examiner.

Applicant's arguments with respect to claims rejection have been fully considered but are moot in view of the new grounds of rejection – see *West et al., OMT, Larsson*, art made of record, as applied hereto.

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Claim Rejections – 35 USC § 102(b)

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102(b) that form the basis for the rejections under this section made in this office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1-11, 19, 21-22, and 27-37 are rejected under 35 U.S.C. 102(b) as being anticipated by West et al., (*The visualization of control logic and physical machine elements within an integrated machine design and control environment*, 2000, Elsevier Science Ltd.) (hereinafter 'West' – art made of record)

3. **As to claim 1** (Currently Amended), West discloses a programming tool for at least one of creating and displaying programs to control a flow of a process using a graphics language for simultaneous representation in a diagram, on a display device interactions of objects that are involved in the control of the process (P. 674, 2nd Par. – elicitation of knowledge from a number of sources in a variety of visual formats [e.g., maps and graphs (to convey relationship amongst parameters and variables), process charts and flowcharts (to describe processes) and physical models, pictures and drawings is required to provide a realistic abstraction] and of a sequence of the interactions of the objects over time (P. 673, 1st Par., Lines 2-11 – the prime motivation for a visual interactive simulation of manufacturing machines is to enable a number of personnel with a variety of backgrounds and interests to actively participate in the

computer modeling of the machine throughout the design and build process lifecycle; the emphasis is in interaction and is fundamentally different from the classical passive animation of systems. Visual interactive simulation is particularly appropriate in the manufacturing machine domain since machines and their constituent components are naturally represented by an iconic visual structure and sequential control logic is mainly deterministic in its operation; Sec. 3.3 – Executing the model and what-if scenarios, 2nd Par. – the control of the animation of the application logic via real-work inputs and state transitions enables time-stamped entries to be recorded by the Synet® Simulator in an event log),

wherein a coordination element is provided (Sec. 3 – Visualization of sequence logic, 1st Par. – manufacturing and process industry control system applications invariably include sequence logic; the complexity of the application logic typically involves the need to manage server concurrent activities, coordinating their behavior to achieve the designed application goal; P. 677, 2nd Par. – Synet® supports a methodology that combines the coordination of the functional approach with the encapsulation of object orientation; P. 694, 2nd Par. – the lines of communication are shown by the lines connecting the objects, i.e., the Conveyor System can only communicate with the Pallet Transfer ...), which establishes (e.g., P. 682, Lines 16-18 – ORBs enable clients to a) locate services, b) activate services if required, c) establish connections and d) enable client-server communication) controls and monitors (e.g., Abstract, 1st Par. – the emphasis changes from obtaining the variation of sensor values and current status of the control logic at the monitoring and operational stages) interactions of the objects

involved and the sequence of the interactions of the objects over time, and wherein the coordination element detects and defines, bases on an interpretation of calls, parameters queries (e.g., Sec. 4 – Visualization of machine elements, 4th Par. – the IMDC-MSM can be used to specify component states, operational parameters, motion parameters and locations; components, parameters and complete sub-assemblies can be stored in the IMDC database) and measurement results (e.g., Abstract, 1st Par. – the information required varies throughout the life cycle of the machine implementation, i.e. the emphasis changes from obtaining results of “what if?” scenarios at the requirements and design states to obtaining the variation of sensor values and current status of the control logic at the monitoring and operational states), which object should be addressed and actuated (e.g., Fig. 2 – Integration of machine modeling (logical and physical) tools, database and target implementation environment using ORB communications within the IMDC environment; Sec. 4 – Visualization of machine elements, 4th Par. – models of physical machines can be graphically constructed with the IMDC-MSM from component building blocks such as axes, sensors, conveyors, actuators, alarms and structural elements).

4. **As to claim 19** (Currently Amended), West discloses a method for programming and representing a program run for at least one of open-loop and closed-loop control of a process, using at least one programmable controller (e.g., Fig. 8 – the integration of individual components within the IMDC environment using the object oriented classification of manufacturing machines and ORB communications; P. 686, 1st Par.), in

which a graphics language is used to implement a process capable of being represented by objects and object interactions, comprising:

- calling a plurality of objects involved in the process in a common diagram (e.g., Fig. 9b – detailed Synect® object sub-hierarchy for Pallet Transfer);
- calling a plurality of respectively required object interactions in the common diagram (e.g., Fig. 9b – detailed Synect® object sub-hierarchy for Pallet Transfer);
- editing the selected objects and object interactions; and translating the previously implemented program into at least one of a corresponding high-level language and a corresponding machine language; the sequence of the object interactions over time in the common diagram (e.g., P. 675, 3rd Par. – Synect® provides a set of software tools (e.g. application editor, compiler, logic engine, logic monitor and code generator) which are integrated into the IMDC platform; Synect® has been designed to allow the logic to be executed and verified by the designer; applications are input by means of a graphical editor);

providing a coordination element (Sec. 3 – Visualization of sequence logic, 1st Par. – manufacturing and process industry control system applications invariably include sequence logic; the complexity of the application logic typically involves the need to manage server concurrent activities, coordinating their behavior to achieve the designed application goal; P. 677, 2nd Par. – Synet® supports a methodology that combines the coordination of the functional approach with the encapsulation of object orientation; P. 694, 2nd Par. – the lines of communication are shown by the lines connecting the

objects, i.e., the Conveyor System can only communicate with the Pallet Transfer ...) that establishes (e.g., P. 682, Lines 16-18 – ORBs enable clients to a) locate services, b) activate services if required, c) establish connections and d) enable client-server communication), controls and monitors (e.g., Abstract, 1st Par. – the emphasis changes from obtaining the variation of sensor values and current status of the control logic at the monitoring and operational stages the interactions of the objects involved and the sequence of the interactions of the objects over time and that detects and defines, based on an interpretation of calls, parameter queries and measurement results (e.g., Sec. 4 – Visualization of machine elements, 4th Par. – the IMDC-MSM can be used to specify component states, operational parameters, motion parameters and locations; components, parameters and complete sub-assemblies can be stored in the IMDC database), which object should be addressed and actuated (e.g., Fig. 2 – Integration of machine modeling (logical and physical) tools, database and target implementation environment using ORB communications within the IMDC environment; Sec. 4 – Visualization of machine elements, 4th Par. – models of physical machines can be graphically constructed with the IMDC-MSM from component building blocks such as axes, sensors, conveyors, actuators, alarms and structural elements); and translating the previously implemented program into at least one of a corresponding high-level language (e.g., Fig. 4 – class diagram for the representations of machine components within the IMDC environment; Sec. 5.1 – IMDC system architecture, 1st Par. – Fig. 4 illustrates a specification class diagram using the Unified Modeling Language (UML) notation for the various class of components that comprise machines with the IMDC

system; P. 682, 1st Par. – the standard architecture enables independent software vendors to develop ORB products that support application interoperability and portability across different programming languages ...; P. 683, 2nd Par. – Language mapping form IDL to C, C++, Ada95 and SmallTalk80 are available) and a corresponding machine language (p. 685, 2nd Par. – these object services, specified using the IDL, define the functions that a tool can perform in response to a request from a remote client and are compiled into the tool).

5. **As to claim 27** (Currently Amended), West discloses a programming tool for creating and providing a graphic representation in a diagram of programs that control the flow of a process, comprising:

- a coordination element (Sec. 3 – Visualization of sequence logic, 1st Par. – manufacturing and process industry control system applications invariably include sequence logic; the complexity of the application logic typically involves the need to manage server concurrent activities, coordinating their behavior to achieve the designed application goal; P. 677, 2nd Par. – Synet® supports a methodology that combines the coordination of the functional approach with the encapsulation of object orientation; P. 694, 2nd Par. – the lines of communication are shown by the lines connecting the objects, i.e., the Conveyor System can only communicate with the Pallet Transfer ...) that established (e.g., P. 682, Lines 16-18 – ORBs enable clients to a) locate services, b) activate services if required, c) establish connections and d) enable client-server communication),

controls and monitors (e.g., Abstract, 1st Par. – the emphasis changes from obtaining the variation of sensor values and current status of the control logic at the monitoring and operational stages) the interactions of objects that are involved in the control of the process and establishes, controls and monitors a sequence of the object interactions over time and that detects and defines, based on an interpretation of calls, parameter queries and measurement results (e.g., Abstract, 1st Par. – the information required varies throughout the life cycle of the machine implementation, i.e. the emphasis changes from obtaining results of “what if?” scenarios at the requirements and design states to obtaining the variation of sensor values and current status of the control logic at the monitoring and operational states),

which object should be addressed and actuated (e.g., Fig. 2 – Integration of machine modeling (logical and physical) tools, database and target implementation environment using ORB communications within the IMDC environment; Sec. 4 – Visualization of machine elements, 4th Par. – models of physical machines can be graphically constructed with the IMDC-MSM from component building blocks such as axes, sensors, conveyors, actuators, alarms and structural elements); and

- a display device that provides a graphic representation of the object interactions together with a graphic representation of the sequence of the object interactions over time in the diagram (P. 673, 1st Par., Lines 2-11 – the prime motivation for a visual interactive simulation of manufacturing machines is to enable a number of

personnel with a variety of backgrounds and interests to actively participate in the computer modeling of the machine throughout the design and build process lifecycle; the emphasis is in interaction and is fundamentally different from the classical passive animation of systems. Visual interactive simulation is particularly appropriate in the manufacturing machine domain since machines and their constituent components are naturally represented by an iconic visual structure and sequential control logic is mainly deterministic in its operation; Sec. 3.3 – Executing the model and what-if scenarios, 2nd Par. – the control of the animation of the application logic via real-work inputs and state transitions enables time-stamped entries to be recorded by the Synet® Simulator in an event log; Fig. 1 – an illustration of the visual interactive simulation process; conceptualization, formalization and exercising and learning).

6. **As to claims 2 (Original), 21 (Original), and 28 (Original),** West discloses a programming tool wherein the process is an automation technology process (e.g., P. 676, 1st Bullet – in the majority of current industrial automation project; P. 695, 3rd Par. – IMDC has the potential to provide a highly effective environment for integrating design and control system elements from a diverse range of manufacturers; the approach gives small and medium sized enterprises the opportunity to participate far more effectively in a vast and growing automation market that is still dominated by large vendors of “closed” systems).

7. **As to claims 3 (Original), 22 (Original), and 29 (Original),** West discloses a method of programming and a programming tool wherein the process is a technical process (e.g., P. 676, 1st Bullet – in the majority of current industrial automation project; P. 695, 3rd Par. – IMDC has the potential to provide a highly effective environment for integrating design and control system elements from a diverse range of manufacturers; the approach gives small and medium sized enterprises the opportunity to participate far more effectively in a vast and growing automation market that is still dominated by large vendors of “closed” systems).

8. **As to claim 4 (Original),** West discloses a programming tool wherein the program is executed in plural, distributed stored program controllers (e.g., P. 672, 3rd par. – the environment is based around a distributed object oriented representation of manufacturing machines as aggregations of basic components: single axes, multi-axes, digital and analogue input/output and dumb and intelligent sensors; distributed object technology provides the core framework within which the tools and real systems intercommunicate).

9. **As to claim 30 (Original),** West discloses a programming tool further comprising plural distributed stored program controllers in which the program is executed (e.g., P. 672, 3rd par. – the environment is based around a distributed object oriented representation of manufacturing machines as aggregations of basic components: single axes, multi-axes, digital and analogue input/output and dumb and intelligent sensors;

distributed object technology provides the core framework within which the tools and real systems intercommunicate).

10. **As to claims 5 (Original) and 31 (Original),** West discloses a programming tool wherein a virtual or additional real processor is provided as the coordination element (e.g., 2nd Par. – the approach taken is unique in that (a) the control logic and physical models of the elements can be investigated individually for correctness and completeness, b) the control logic can be easily integrated with the physical models to animate the modeled physical machine and c) reconfiguration enables the same control logic to be applied to real world physical control elements).

11. **As to claims 6 (Original) and 32 (Original),** West discloses a programming tool wherein a virtual processor or an additional real processor is provided as the coordination element in connection with the distributed stored program controllers (e.g., Sec. 5.1 – IMDC system architecture, 2nd Par. – clients and servers have integrated using an integrating infrastructure based upon the Common Object Request Broker Architecture (CORBA) specification for Object Request Brokers (ORBs) from the Object management Group (OMG); the Distributed Object Management Environment (DOME) from Object Technology® ORB product was adopted for the IMDC infrastructure due to the compatibility with real time operating system).

12. **As to claim 7 (Original) and 33 (Original),** West does not disclose a programming tool wherein at least substantially all calls of the objects are processed by the coordination element (e.g., Sec. 3 – Visualization of sequence logic, 1st Par. – manufacturing and process industry control system applications invariably include sequence logic; the complexity of the application logic typically involves the need to manage server concurrent activities, coordinating their behavior to achieve the designed application goal; P. 677, 2nd Par. – Synet® supports a methodology that combines the coordination of the functional approach with the encapsulation of object orientation; P. 694, 2nd Par. – the lines of communication are shown by the lines connecting the objects, i.e., the Conveyor System can only communicate with the Pallet Transfer ...).

13. **As to claims 8 (Original) and 34 (Original),** West discloses a programming tool wherein the coordination element determines at least one of the instant of each call and the addressee of each call (e.g., P. 686, 1st par. – two object instance of an Axis-Controller calls can be create...).

14. **As to claim 9 (Original),** West discloses a programming tool wherein the graphics language comprises a graphic representation of all of the objects and a graphic representation of all of the object interactions, wherein each graphic representation, of the objects and the object interactions, respectively, is called and interconnected using an editor to implement an executable program (e.g., P. 675, 3rd Par. – Synect® provides a set of software tools (e.g. application editor, compiler, logic

engine, logic monitor and code generator) which are integrated into the IMDC platform; Synect® has been designed to allow the logic to be executed and verified by the designer; applications are input by means of a graphical editor).

15. **As to claim 35** (Original), West discloses a programming tool further comprising an editor that calls and interconnects the graphic representation of the objects and the graphic representation of the object interactions, respectively, to implement an executable program (e.g., P. 675, 3rd Par. – Synect® provides a set of software tools (e.g. application editor, compiler, logic engine, logic monitor and code generator) which are integrated into the IMDC platform; Synect® has been designed to allow the logic to be executed and verified by the designer; applications are input by means of a graphical editor).

16. **As to claims 10 (Original) and 36 (Original)**, West discloses a programming tool wherein each graphic representation in the diagram of an object and an object interaction is associated with an instruction or a program module (e.g., Fig. 4 – class diagram for the representations of machine components within the IMDC environment).

17. **As to claims 11 (Original) and 37 (Original)**, West discloses a programming tool wherein the instruction or the program module is in machine language (e.g., p. 685, 2nd Par. – these object services, specified using the IDL, define the functions that a tool can perform in response to a request from a remote client and are compiled into the tool).

Claim Rejections – 35 USC § 103(a)

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

18. Claims 12 and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over West in view of OMG, (*Object Groups: A response to the ORB 2.0 RFI, 1993, The Object management Group (OMG)* (hereinafter 'OMG' – art made of record)

19. **As to claims 12 (Original) and 38 (Original),** West does not explicitly disclose a programming tool wherein the following additional object interactions: branching of an object call; parallel connection of an object call; synchronized connection of at least two interactions loop or jump to repeat at least one of an instruction and a program segment; are each represented conditionally or unconditionally in the diagram and are thereby implemented correspondingly.

However, in an analogous art of Object Groups: A response to the ORB 2.0 RFI, OMG discloses a programming tool wherein the following additional object interactions: branching of an object call; parallel connection of an object call (e.g., P. 3, sub-sec. – parallelism and load sharing); synchronized connection (e.g., P. 4, 2nd Par. – the same

protocols used to obtain consistency in data notification applications also underly so-called virtually synchronous master-slave designs in which slave objects coordinate solution of subtasks without the need for application programmers to arrange complex, fragile, and often incorrect concurrency control measures; P. 11, sub-sec. – FIFO ordering) of at least two interactions loop or jump to repeat at least one of an instruction and a program segment; are each represented conditionally or unconditionally (P. 9, 1st Par.) in the diagram and are thereby implemented correspondingly.

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of OMG into the West's system to further a programming tool wherein the following additional object interactions: branching of an object call; parallel connection of an object call; synchronized connection of at least two interactions loop or jump to repeat at least one of an instruction and a program segment; are each represented conditionally or unconditionally in the diagram and are thereby implemented correspondingly in West system.

The motivation is that it would further enhance the West's system by taking, advancing and/or incorporating OMP's system which offers significant advantages that the object group framework couples the benefits of object oriented programming and Isis technology to permit the design, implementation and deployment of complex distributed application from independent, reusable components as once suggested by OMP (e.g., P. 2, 2nd Par.).

20. Claims 13-18, 20-26, and 39-43 are rejected under 35 U.S.C. 103(a) as being unpatentable over West in view of Tony Ingemar Larsson, (Pat. No. US 6,467,085 B2) (hereinafter 'Larsson' – art made of record)

21. **As to claims 13 (Original) and 39 (Original),** West discloses a graphic editor (e.g., Sec. 3.3 – Executing the model and what-if scenarios, 1st Par.– A Synect™ Monitor application enables the application's current state, enabled transitions and status of real-world inputs to be presented to the users by visual animation of the designer's specification with the Synect™ Editor tool; Fig. 11 – Typical user interface for the Synect™ logic editor), but does not explicitly disclose a programming tool wherein a representation of the graphics language in the diagram shows the object interactions a first axis, and shows a sequence of the object interactions over time on a second axis of the diagram.

However, in an analogous art of System and Method for Reducing Coupling in an Object-Oriented Programming Environment, Larsson discloses a programming tool wherein a representation of the graphics language in the diagram shows the object interactions a first axis, and shows a sequence of the object interactions over time on a second axis of the diagram (e.g., Fig. 2; Col. 5, Lines 13-21 – this is often done using an inter-working diagram, as shown in Fig. 2; An inter-working diagram is also referred to by those skilled in the art as object-interaction sequence, message-passing sequence or timing diagram; sometimes, such as diagram is rotated 90° so that the time-axis becomes vertical).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Larsson into the West's system to further provide a programming tool wherein a representation of the graphics language in the diagram shows the object interactions a first axis, and shows a sequence of the object interactions over time on a second axis of the diagram in West system.

The motivation is that it would further enhance the West's system by taking, advancing and/or incorporating Larsson's system which offers significant advantages that all object interactions are indirect, via interface objects such as a centralized switch object, through an agent object, or an object responding like either a centralized switch object or an agent object as once suggested by Larsson (e.g., Col. 7, Lines 60-63).

22. **As to claim 14** (Original), West discloses a programming tool wherein the representation of the graphics language in the diagram is real-time capable (e.g., P. 681, 3rd Bullet – the Distributed Runtime Machines).

23. **As to claims 15 (Original) and 43 (Original)**, West discloses a programming tool wherein the display device is associated with a buffer memory for buffered representation of the flow of the process using the graphics language (e.g., P. 676, 1st Par. – the combinational logic approach is potentially the preferred option if the application logic is very simple or there is a strong need for the solution to require the minimum of memory or execute as fast as possible).

24. **As to claims 16 (Original) and 44 (Original),** West discloses a programming tool wherein a sequence chart representation is selected as the diagram (Sec. 3 – Visualization of sequence logic, 1st Par. – manufacturing and process industry control system applications invariably include sequence logic; the complexity of the application logic typically involves the need to manage server concurrent activities, coordinating their behavior to achieve the designed application goal; P. 677, 2nd Par. – Synet® supports a methodology that combines the coordination of the functional approach with the encapsulation of object orientation; P. 694, 2nd Par. – the lines of communication are shown by the lines connecting the objects, i.e., the Conveyor System can only communicate with the Pallet Transfer ...).

25. **As to claims 17 (Original) and 40 (Original),** Larsson discloses a programming tool wherein the diagram shows the sequence of object interactions over time on the second axis of the diagram from top to bottom (e.g., Fig. 2; Col. 5, Lines 13-21 – this is often done using an inter-working diagram, as shown in Fig. 2; An inter-working diagram is also referred to by those skilled in the art as object-interaction sequence, message-passing sequence or timing diagram; sometimes, such as diagram is rotated 90° so that the time-axis becomes vertical)

26. **As to claims 18 (Original), 41 (Original), and 42 (Original),** West discloses a programming tool wherein the graphics language in the diagram can be constructed in

real time (e.g., P. 681, 3rd Bullet – the Distributed Runtime Machines; P. 682, 1st Par. – the Distributed Object Management Environment (DOME) from Object Technology ORB product was adopted for the IMDC infrastructure due to the compatibility with real time operation system).

27. **As to claim 20** (Original), West discloses a graphic editor (e.g., Sec. 3.3 – Executing the model and what-if scenarios, 1st Par.– A Synect™ Monitor application enables the application's current state, enabled transitions and status of real-world inputs to be presented to the users by visual animation of the designer's specification with the Synect™ Editor tool; Fig. 11 – Typical user interface for the Synect™ logic editor), but does not explicitly disclose a method wherein the objects and the object interactions are arranged on a first axis of the common diagram, and wherein the successive sequence of the object interactions over time is represented by arranging the object interactions on a second axis of the common diagram.

However, in an analogous art of System and Method for Reducing Coupling in an Object-Oriented Programming Environment, Larsson discloses a method wherein the objects and the object interactions are arranged on a first axis of the common diagram, and wherein the successive sequence of the object interactions over time is represented by arranging the object interactions on a second axis of the common diagram (e.g., Fig. 2; Col. 5, Lines 13-21 – this is often done using an inter-working diagram, as shown in Fig. 2; An inter-working diagram is also referred to by those skilled in the art as object-interaction sequence, message-passing sequence or timing

diagram; sometimes, such as diagram is rotated 90° so that the time-axis becomes vertical).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Larsson into the West's system to further provide a method wherein the objects and the object interactions are arranged on a first axis of the common diagram, and wherein the successive sequence of the object interactions over time is represented by arranging the object interactions on a second axis of the common diagram in West system.

The motivation is that it would further enhance the West's system by taking, advancing and/or incorporating Larsson's system which offers significant advantages that all object interactions are indirect, via interface objects such as a centralized switch object, through an agent object, or an object responding like either a centralized switch object or an agent object as once suggested by Larsson (e.g., Col. 7, Lines 60-63).

28. **As to claim 23** (Original), West discloses a method for programming wherein at least one of the arrangement of the objects and the object interactions, and the representation of the successive sequence of the object interactions over time, in the common diagram are real-time capable (e.g., P. 681, 3rd Bullet – the Distributed Runtime Machines; P. 682, 1st Par. – the Distributed Object Management Environment (DOME) from Object Technology ORB product was adopted for the IMDC infrastructure due to the compatibility with real time operation system).

29. **As to claim 24** (Original), Larsson discloses a method for programming wherein a sequence chart representation is selected as the common diagram (e.g., Fig. 2; Col. 5, Lines 13-21 – this is often done using an inter-working diagram, as shown in Fig. 2; An inter-working diagram is also referred to by those skilled in the art as object-interaction sequence, message-passing sequence or timing diagram; sometimes, such as diagram is rotated 90° so that the time-axis becomes vertical).

30. **As to claim 25** (Original), Larsson discloses a method for programming wherein the common diagram is two-dimensional (e.g., Fig. 2; Col. 5, Lines 13-21 – this is often done using an inter-working diagram, as shown in Fig. 2; An inter-working diagram is also referred to by those skilled in the art as object-interaction sequence, message-passing sequence or timing diagram; sometimes, such as diagram is rotated 90° so that the time-axis becomes vertical).

31. **As to claim 26** (Original), Larsson discloses a method for programming wherein the successive sequence of the object interactions over time is represented by arranging the object interactions from top to bottom on the second axis of the common diagram (e.g., Fig. 2; Col. 5, Lines 13-21 – this is often done using an inter-working diagram, as shown in Fig. 2; An inter-working diagram is also referred to by those skilled in the art as object-interaction sequence, message-passing sequence or timing diagram; sometimes, such as diagram is rotated 90° so that the time-axis becomes vertical).

Conclusion

32. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ben C. Wang whose telephone number is 571-270-1240. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.



TUAN DAM
SUPERVISORY PATENT EXAMINER

BCW 
August 7, 2007